PTO/SB/21 (09-04)
Approved for use through 07/31/2006 OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission: 15

Application Number	09/739,516
Filing Date	12/18/2000
First Named Inventor	LEWIS
Art Unit	2194
Examiner Name	L. TRUONG
Attorney Docket Number	PIT-US000345

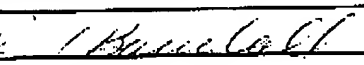
ENCLOSURES *(Check all that apply)*

- ☐ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment/Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/ Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation
- ☐ Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) ____
  - ☐ Landscape Table on CD
- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☑ Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) (please identify below):

Remarks

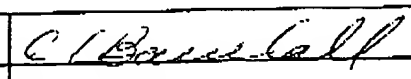
PLEASE CHARGE ANY FEES OTHER THAN THE ISSUE FEE TO DEPOSIT ACCOUNT 14-1270. PLEASE CREDIT ANY OVERPAYMENTS TO THE SAME ACCOUNT.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	ANNE E. BARSCHALL, ATTORNEY		
Signature			
Printed name	ANNE E. BARSCHALL		
Date	11/14/05	Reg. No.	31,089

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	ANNE E. BARSCHALL	Date	11/14/05

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application Ser. No.: 09/739,516 Group Art Unit: 2194

Filing Date: 12/18/2000 Examiner: L. TRUONG

Attorney Docket Number US 000345 Inventor Name(s): LEWIS

Confirmation # 3122

Title: SELF-DETERMINING COMMAND PATH ARCHITECTURE

Mail Stop Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

APPEAL BRIEF

Sir:

This is an appeal from the final rejection of Claims 1-3, 6-7, and 10-13.

I. REAL PARTY IN INTEREST

The real party in interest is Koninklijke Philips Electronics, N.V., a corporation of the Netherlands.

II. RELATED APPEALS AND INTERFERENCES

Applicant is not aware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-3, 6, 7, and 10-13 stand rejected. Claims 4-5 and 8-9 have been cancelled or withdrawn from consideration[1]. Claims 14-18 were not entered.

IV. STATUS OF AMENDMENTS

The amendment under rule 116 was not entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The invention relates to the field of object oriented programming.[2] The invention relates to governing the behavior of objects in a data processing system. In the preferred embodiment, per Fig. 2, a data object 110 is put into a first queue 141, based on the contents of a path object 115. Once queued the data object 110 is processed by processing object 1 at 100. The

---

[1] The status of claims 4-5 is a bit unclear. Applicant attempted to cancel them in the last amendment under rule 116. That amendment was not entered. However the advisory action indicates that these claims were nevertheless withdrawn from consideration. Applicant therefore is assuming that the claims were cancelled by Examiner's amendment or that the rule 116 was partially entered.

[2] The following is copied from the prior appeal brief in case the board is not familiar with the field.

The application contains extensive background as to what object-oriented programming is on pages 1-5. The application also defines the term "object" especially on page 2, lines 12 et seq.

Each object within an O-O system is defined by an interface and an implementation. A software client external to an object depends completely on its interface and not the details of its implementation. The implementation of an object provides the mechanisms and the details that define its behavior. O-O programs are collections of objects that relate to each other through their interfaces.

In a sense, each object is a "black box." Its interface consists of messages that the black box sends and receives. *Objects actually contain code (sequences of computer instructions) and data (information which the instructions operate on).* Traditionally, code and data have been kept apart. For example, in the C language, units of code are called "functions," while units of data are called "structures." Functions and structures are not formally connected in C. A C function can operate on more than one type of structure, and more than one function can operate on the same structure. This is not true for O-O software. In O-O programming, code and data are merged into a single indivisible thing – an object. A programmer using an object should not need to look at the internals of the object once the object has been defined. *All connections with the object's internal programming are accomplished via messages; i.e., the object's interface.* [emphasis added]

It can be seen from this definition that an object is not the same as data or a data structure. A data structure is something that contains data to be used by code. The object can contain data structures, but those data structures will only be accessible to that object using that object's code. Data structures, unlike objects, typically do not send messages. Data structures are typically passive.

It can also be seen from this definition that an object is not the same as a piece of hardware. An object is data and code, independent of particular hardware implementations. The fact that an object is represented as a block in a figure does not make it a hardware device. Thus processing objects are not processors. Processors are hardware devices.

processing object 1 return a processing status to the path object 115. If the status is normal 181, the object is passed to queue 141 and thence to processing object 2 at 120. If the status is error 182, then the data object 110 is passed to queue 143 and thence to processing object 3 at 130.

Path Object (Independent claims 1, 6, 12, and 13)

The term "path object" is defined in the specification at p.5, ll. 21-. "*The path object itself is an organization of the available queues in the system.*" This is illustrated with respect to the preferred embodiment at 212 in Fig. 3, where each of "que" 1 through n has the structure "QueStruct." This path object, as defined in the specification and recited in all the independent claims, results in the functional advantage of allowing the data object to be routed through numerous nodes based on path information contained in or associated with the data object. The system does not have to keep track of the data object's path independent of the data object.

Claim 10

Claim 10 recites that the path object includes a table of queue indicators (spec. p. 9, line 12). Claim 10 depends from claim 1. Incorporated claim 1 also recites that the path object is associated with a respective data object (p. 9, lines 11-12).

Claim 11

Claim 11 depends from claim 1 and recites that the processing comprises determining a normal or faulty outcome, spec. p. 9, lines 13-14. The processing, as recited in claim 1, is of a data object in a processing object, spec. p. 9, lines 10-11. The claim further recites identifying,

namely identifying a queue, in response to the outcome, spec. p. 9, line 17.

Claim 12

The preferred embodiment of this claim is discussed in the specification at page 9, line 8, et seq. with respect to Fig. 2. Claim 12 recites that the first queuing (line 9) is in response to an indication of the first processing object (line 10, Fig. 2, 100) in a path object (Fig. 2, 115) ; and that the second queuing (line 18) is in response to both results of the processing (lines 13-14) and an indication (line 17) of the second processing object (lines 21-22, Fig. 2, 120) in the path object (Fig. 2, 115). In other words, the path object defines at least two segments of a path for the data object.

VI. GROUNDS OF RJECTION TO BE REVIEWED ON APPEAL

Section 112 rejections of claims 12 and 13.

Art rejections of claims 1-3, 6-7, and 10-13.

VII. THE ARGUMENT

Chang/Winnick combination (Claim 1)

Winnick and Chang are from different fields. Chang relates to an object-oriented software device, while Winnick relates to conventional hardware devices. Applicant respectfully submits that one of ordinary skill in the art would not look to Winnick to supplement the teachings of Chang. Applicant accordingly respectfully submits that the combination of these references is an improper hindsight reconstruction.

Path object (Claims 1, 6, 12, and 13)

Against the recitation of "path object," per claims 1, 6, 12, and 13, the Examiner cites Chang, col. 13, ln 15-30; col. 4, ll 30-34; col. 17, ll. 65-67, and col. 18, ll 1-3, particularly with respect to the Network_ContextID object and a Service_ContextID object. Applicant has reviewed Chang and does not believe that the cited sections teach or suggest a path object as defined by Applicant; nor, so far as Applicant can tell, does the reference teach or suggest the functional advantages of Applicant's path objects.

First, there appear to be two objects in question, not a path object as recited.

Moreover, the cited portions of Chang teach information about transferring messages along a single, current path segment. Applicant does not find any teaching or suggestion that either Network_ContextID or Service_ContextID is a "path object," as defined in the specification, i.e. they fail to teach or suggest an organization of available queues in the system.

Applicant accordingly respectfully submits that the Examiner has not presented a *prima facie* case against the claims.

Claim 12

The Examiner incorrectly states that claim 12 is the same as claims 1, 4 and 5. These claims are all too different from each other to be grouped this way.

Claims 4 and 5 have been cancelled. They did not, in any case, recite the same invention as claim 12. For instance, they recited modifying an indicator portion of an object to create a new object. This recitation does not appear in claim 12. Because of the claim differences,

Applicant submits that incorporating the rejections of claims 4-5 into the rejection of claim 12 is confusing and not in conformity with 37 CFR 1.104.

Claim 12 also recites first and second queuing. Both queuings are in response to an indication in the path object. Therefore the path object has queue in formation for more than one process object. This is just exemplary of recitations that Claim 1 does not share with claim 12. Applicant therefore submits that incorporating the rejection of claim 1 into the rejection of claim 12 is also confusing and not in conformity with 37 CFR 1.104.

Against claim 12, the Examiner cites Chang as previously applied. So far as Applicant can tell from the portions of Chang cited by the Examiner, Chang only indicates source and destination for a single segment of a path — using the Network_ContextID and Service_ContextID objects. In other words, taking an object from one place to a queue of a process. Applicant does not find that there is teaching or suggestion of the network_contextID and service_contextID objects containing queue information for more than one process. They do not appear to be able to support first and second queuing. Applicant accordingly respectfully submits that the Examiner has failed to make a *prima facie* case against claim 12.

Claim 13

Applicant respectfully submits that the Examiner improperly groups claims 13 and 12 together, when their recitations are totally different. Accordingly, the Examiner has failed to meet the requirements of 37 CFR section 1.104 with respect to claim 13.

Winnick (claim 1)

Claim 1 recites identifying a queue based on an indicator in the path object. Against this recitation, the Examiner cites Winnick at col. 1, ll. 58-62.

Applicant respectfully submits that the Examiner mischaracterizes Winnick. The messages in the cited portion of Winnick do not appear to be data objects. They do have status indicators, but, so far as Applicant can tell by reading the paragraph in which this section appears, these indicators are not used to identify a queue, unlike the recitations of claim 1. Instead, these status indicators are used to indicate whether the transmitter is in active or standby mode.

Applicant suspects that the Winnick reference was uncovered during a keyword search on the word "indicator." "Indicator" is a common English word that appears in many patents and has many different meanings, depending on context. The presence of this word, without more, does not indicate that the patent is relevant art. Applicant respectfully submits that keyword searches of this type are improper hindsight reconstructions that use Applicant's claims as a road map. The CAFC has said

> The "as a whole" instruction in title 35 prevents evaluation of the invention part by part. Without this important requirement, an obviousness assessment might break an invention into its component parts (A + B + C), then find a prior art reference containing A, another containing B, and another containing C, and on that basis alone declare the invention obvious. This form of hindsight reasoning, using the invention as a roadmap to find its prior art components, would discount the value of combining various existing features way to achieve a new result – often the very definition of invention. Ruiz v. A. B. Chance Co., http://www.ll.georgetown.edu/federal/judicial/fed/opinions/03opinions/03-

1333.html at p. 7; 357 F. 3d 1270; 2004 US App. Lexis 1325; 69 U.S.P.Q. 2d (BNA) 1686 (Fed. Cir 2004)

Applicant therefore respectfully submits that the Examiner has failed to make a *prima facie* case against claim 1.

Claim 10

Claim 10 recites that the path object includes a table of queue indicators. Claim 10 depends from claim 1. Claim 1 also recites that the path object is associated with a respective data object.

Against this recitation, the Examiner cites Nakamura. Again, like Winnick, Nakamura appears to be a hardware system with messages between hardware units. Accordingly, one of ordinary skill in the art would not apply it to an O-O software context like Chang. Applicant therefore respectfully submits that the combination is improper.

Moreover, so far as Applicant can tell, by reading the cited portions of the Nakamura reference, the destination registration table 40 is not a path object associated with a data object. Instead, it appears that this table is used to keep track of the status of destinations for the system as a whole, not to hold data with respect to paths for particular data.

Again this rejection appears to result from a mere keyword search on the word "table," which has produced a piece of irrelevant art that happened to have the same word as Applicant's claim. "Table" is a common English word that appears in many patents and has many different meanings, depending on context. Applicant respectfully submits that this is an improper hindsight reconstruction using Applicant's claims as a road map.

Claim 11

Against the recitations of this claim, the Examiner cites Winnick bottom of col. 1, though perhaps he also means the top of col. 2. Applicant has reviewed the cited portion of the references and respectfully submits that the Examiner mischaracterizes Winnick in this respect. Applicant understands Winnick to transmit an indication of the status of the transmitter, whether standby or active. Applicant does not understand any queues to be identified for data in accordance with the active or standby mode or in response to any system faults. Applicant therefore concludes that this portion of Winnick does not stand for the proposition for which the Examiner cites it. Applicant therefore respectfully submits that the Examiner has failed to make a *prima facie* case of obviousness against claim 11.

Section 112 rejection (claims 12 & 13)

The Examiner seems to think that claims 12 and 13 recite first and second data objects, like claim 4. Applicant does not find this recitation in claims 12 and 13. Applicant accordingly respectfully submits that the Examiner has improperly grouped claims 12 and 13 with claim 4 in this rejection. Withdrawal of the rejection against claims 12 and 13 is accordingly respectfully requested.

VIII. CONCLUSION

Applicant respectfully submits that he has answered each issue raised by the Examiner. Withdrawal of the finality of the last office action and entry of the amendment under rule 116 are therefore respectfully requested.

Respectfully submitted,

By _____

Anne E. Barschall
Reg. No. 31,089
(914) 332-1019
fax 914-332-7719
November 14, 2005

CLAIMS APPENDIX

1. (previously presented) A method of determining the flow of a data object in a software architecture using queues to organize the transfer of data from one processing object to another, comprising :

storing a queue indicator in a path object corresponding to a respective data object;

receiving and processing the data object in a first of said processing objects;

identifying a queue corresponding to a second of said processing objects the identifying depending on the indicator in the path object corresponding to said data object;

placing said data object in the queue identified in said step of identifying.

2. (previously presented) A method as in claim 1, wherein said step of identifying includes determining a result of said step of processing.

3. (previously presented) A method as in claim 1, wherein: said step of identifying includes determining a result of said step of processing; and said queue corresponding to said result.

4-5. (cancelled or withdrawn[3])

6. (previously presented) A pipeline software architecture in which data objects are transferred

[3] Please see the footnote in the "status of claims" section of this brief.

from a first processing object to a selected one of second and third processing objects by queuing the data objects in a queue of said selected one, comprising:

a path object corresponding to each of said data objects;

at least one of said path objects containing an indicator of at least one of said second and third processing object;

said first processing object defining a process a result of which is to insure that a first data object processed by said first processing object is placed in a queue of said at least one of said second and third processing objects responsively to one of said path objects corresponding to said first data object.

7. (previously presented) An architecture as in claim 6, wherein said process includes the generation of an indication of a result of processing of said first processing object and said first data object processed by said first processing object is placed in said queue of said at least one of said second and third processing objects responsively to the processing object indicator in the at least one of said path objects corresponding to said first data object and responsively to said result indication.

8-9. (canceled)

10. (previously presented) The method of claim 1, wherein the path object includes a table of queue indicators.

11. (previously presented) The method of claim 1, wherein

the processing comprises determining a normal or faulty outcome state of the data object; and

the identifying is dependent on said normal or faulty outcome state.

12. (previously presented) A method comprising:

defining objects, each comprising both data and functions that access the data, the objects including: data objects, and path objects and processing objects;

first queuing a data object in a queue of a first processing object in response to a [4]indication of the first processing object in a path object associated with the data object;

responsive to the first queuing, processing the data object with the first processing object;

second queuing the data object in a queue of a second processing object in response to both: results of the processing; and an indication of the second processing object in the path object associated with the data object;

responsive to the second queuing, processing the data object with a second processing object.

13. (previously presented) Apparatus comprising:

objects, each object comprising both data and functions that access the data, the objects including: data objects and path objects and processing objects, each path object mutually

[4] sic, should be "an."

corresponding to a respective data object;

a respective processing queue for each processing object, the processing objects each process each data object previously queued in the respective queue, the processing of the data object including using the functions of the data object to access the data of the data object, the path objects each comprising indicators of next processing objects for subsequent processing of the corresponding data object after the processing of the data object by the current processing object, the current processing object communicating with the path object to determine the next processing objects for subsequent processing[5] the data object, after the processing of the data object by the current processing unit is complete the processing object queues the data object in a queue of one of the next processing objects depending on a result of the processing of the data object by the current processing object.

[5] sic, should insert "of"